

Tecnologias e Programação de Sistemas de Informação

Android

Arquitetura de Dispositivos | David Jardim

Cofinanciado por:



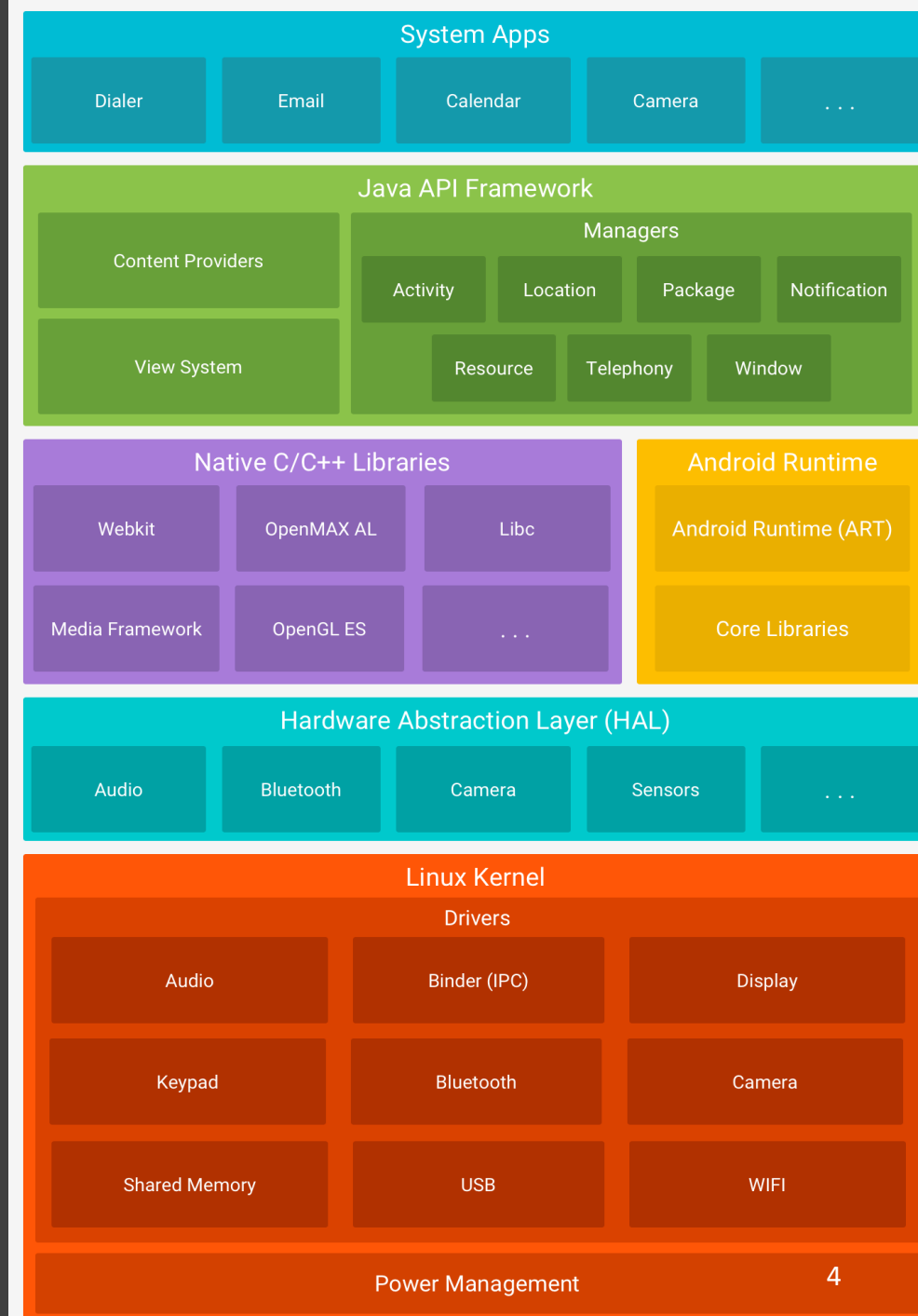
Da aula anterior...

- Excepções
- Java SWING
 - Componentes
 - Eventos
 - *Listeners*

Android

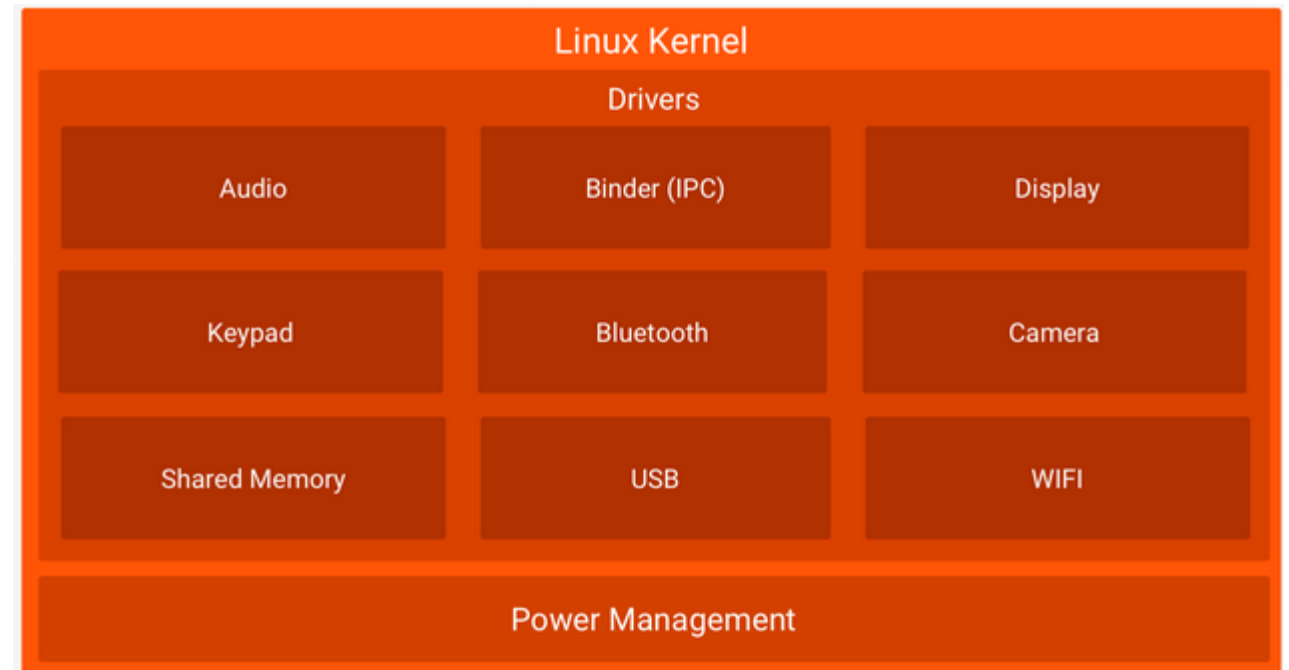
- Sistema operativo baseado em Linux
- Desenhado principalmente para dispositivos móveis

Arquitetura da Plataforma



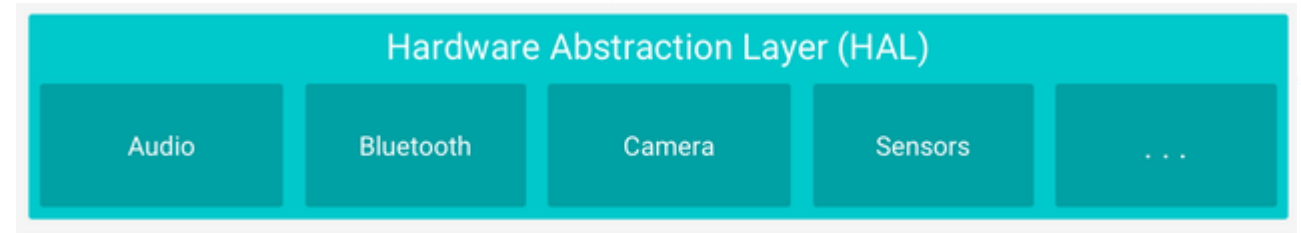
Kernel Linux

- Gestão de memória e processamento
- Permite aos fabricantes o desenvolvimento e utilização de drivers
- Providencia comunicações seguras entre aplicações



Hardware Abstraction Layer

- Fornece interface entre o hardware e a framework Java API
- Composta por diversos módulos
- Permite aceder a um determinado componente, como a câmara, ou microfone



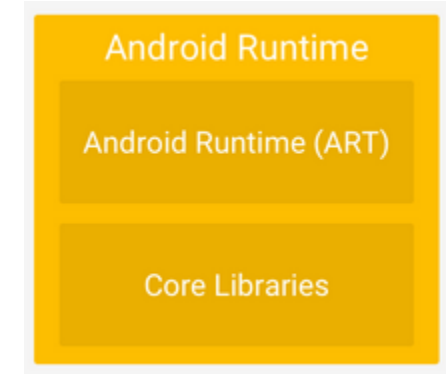
Native C/C++ Libraries

- Alguns serviços nativos são escritos em C/C++
- Esses serviços são expostos através da Java framework API
- No entanto é possível desenvolver em C/C++ utilizando o toolset Android NDK



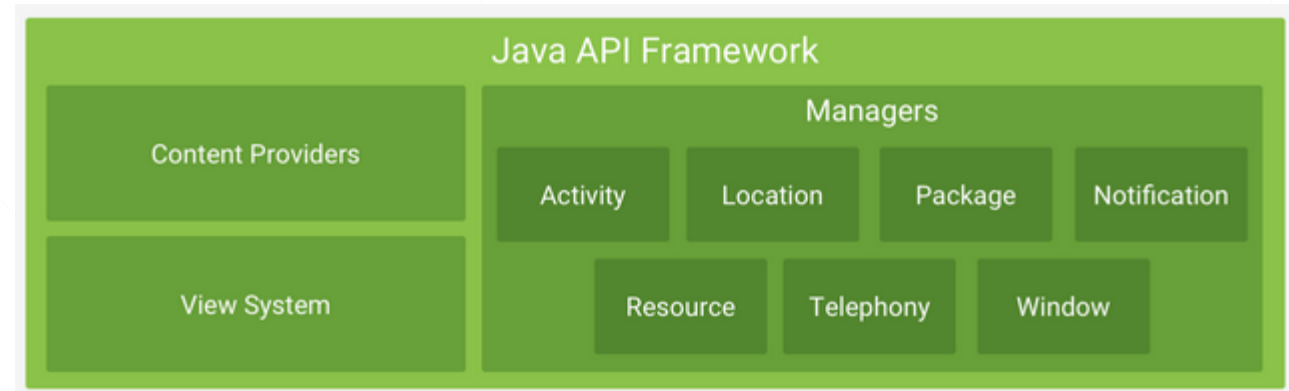
Android Runtime

- Responsável pela execução das aplicações
- Cada aplicação corre numa instância do ART>
- Responsável pelo Garbage Collector
- Permite efetuar debugging, profiling, diagnósticos e relatórios



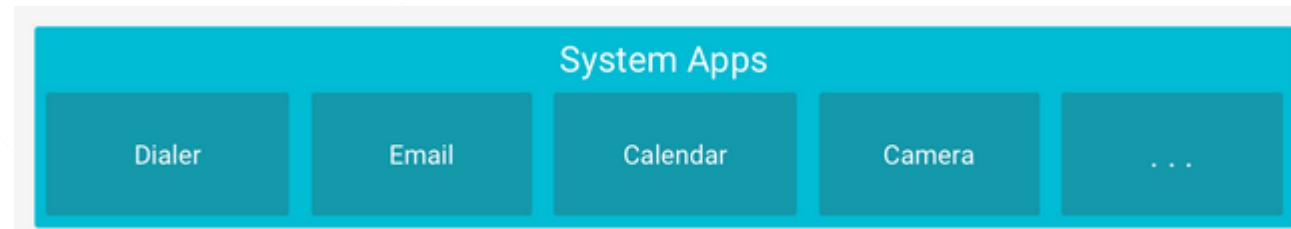
Java API Framework

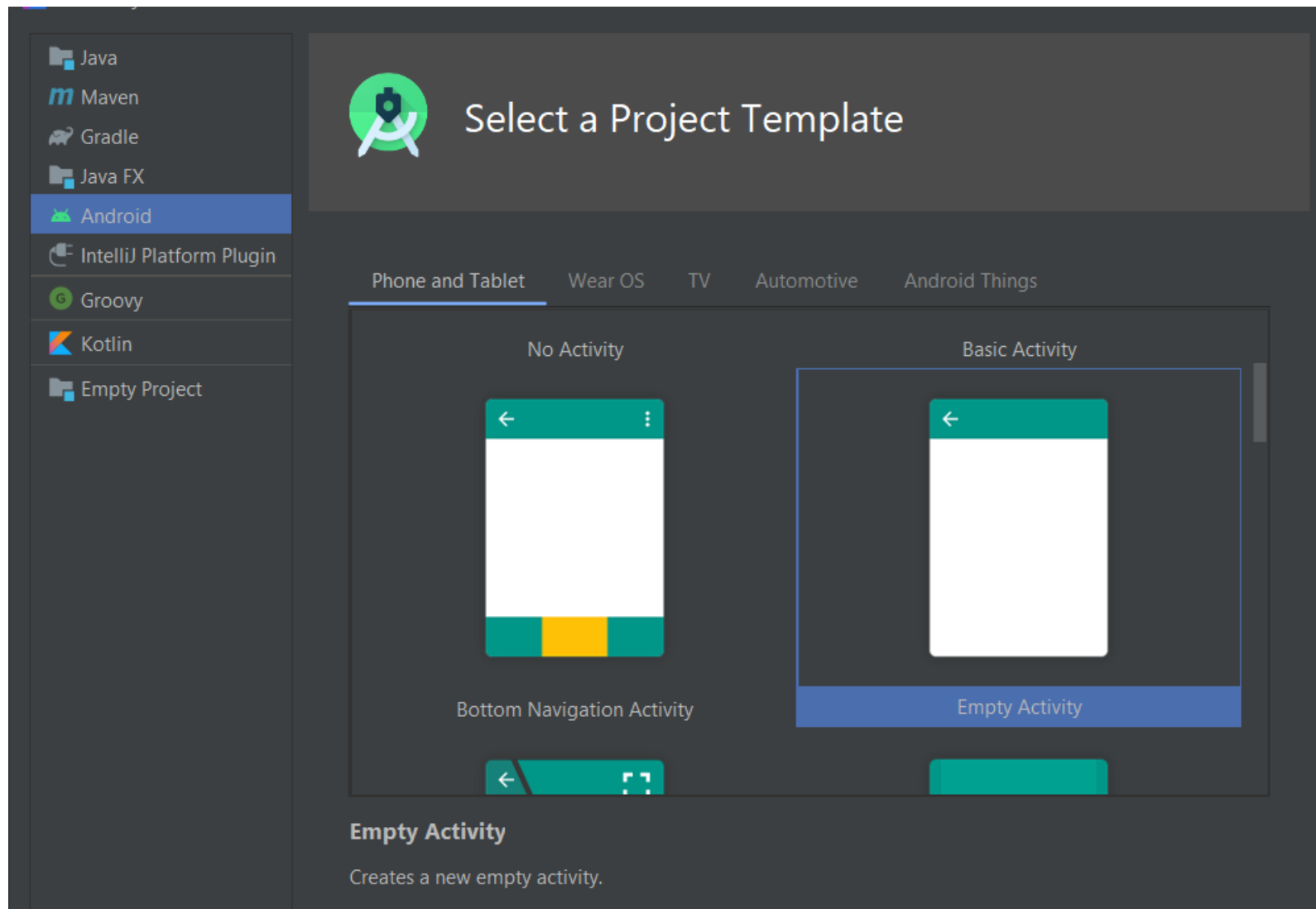
- Content Providers permitem a partilha de dados entre aplicações
- O View System permite a criação de interfaces gráficas ricas
- Disponibiliza todas as funcionalidades do SO através de APIs escritas em Java
- Essas funcionalidades são disponibilizadas através de um conjunto de managers



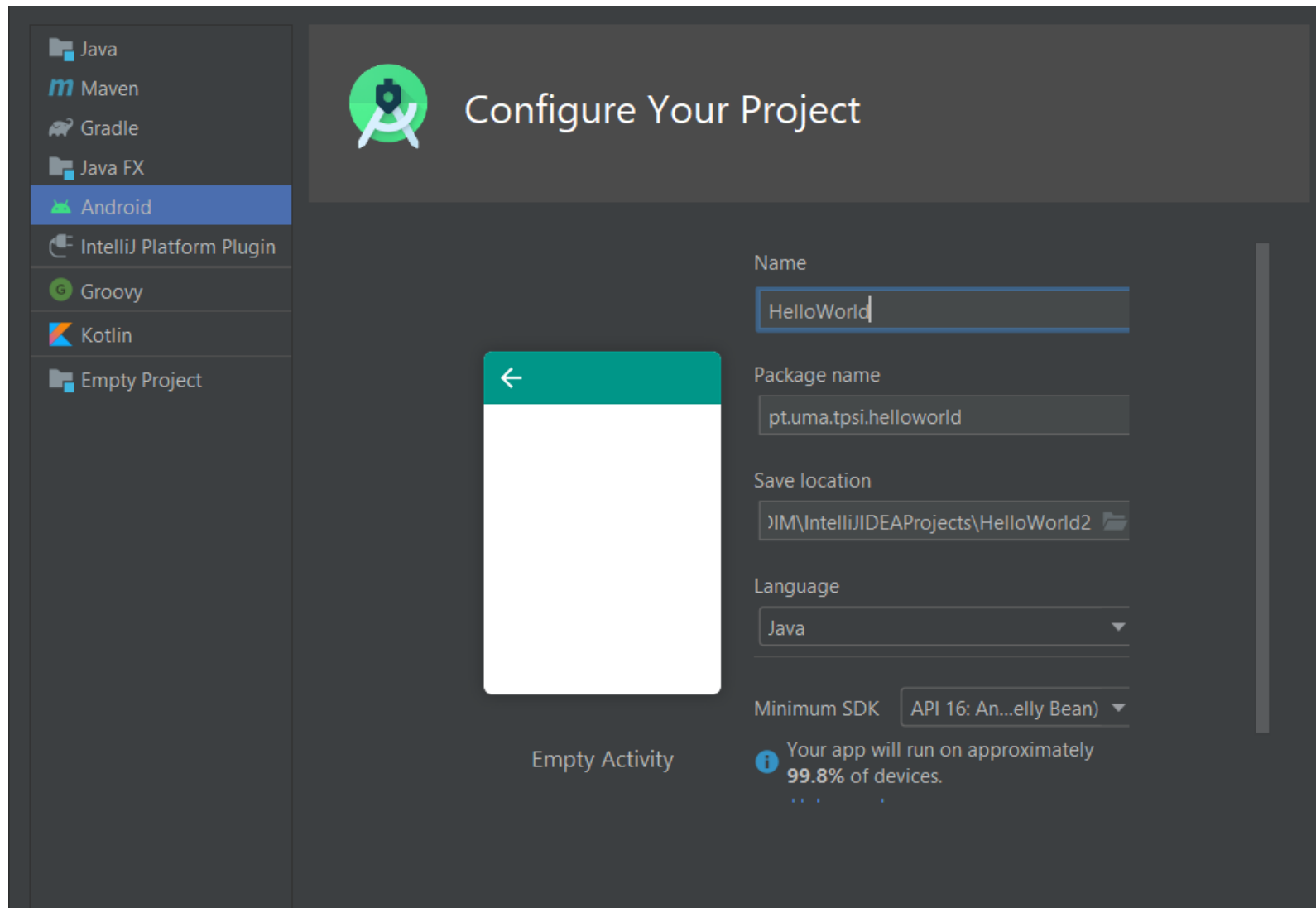
System Apps

- Conjunto default de aplicações
- SMS, email, calendário, browser, contatos, chamadas, etc
- Estas aplicações podem ser invocadas a partir de aplicações desenvolvidas por nós
- Por exemplo, enviar uma SMS quando ocorrer algo na nossa APP



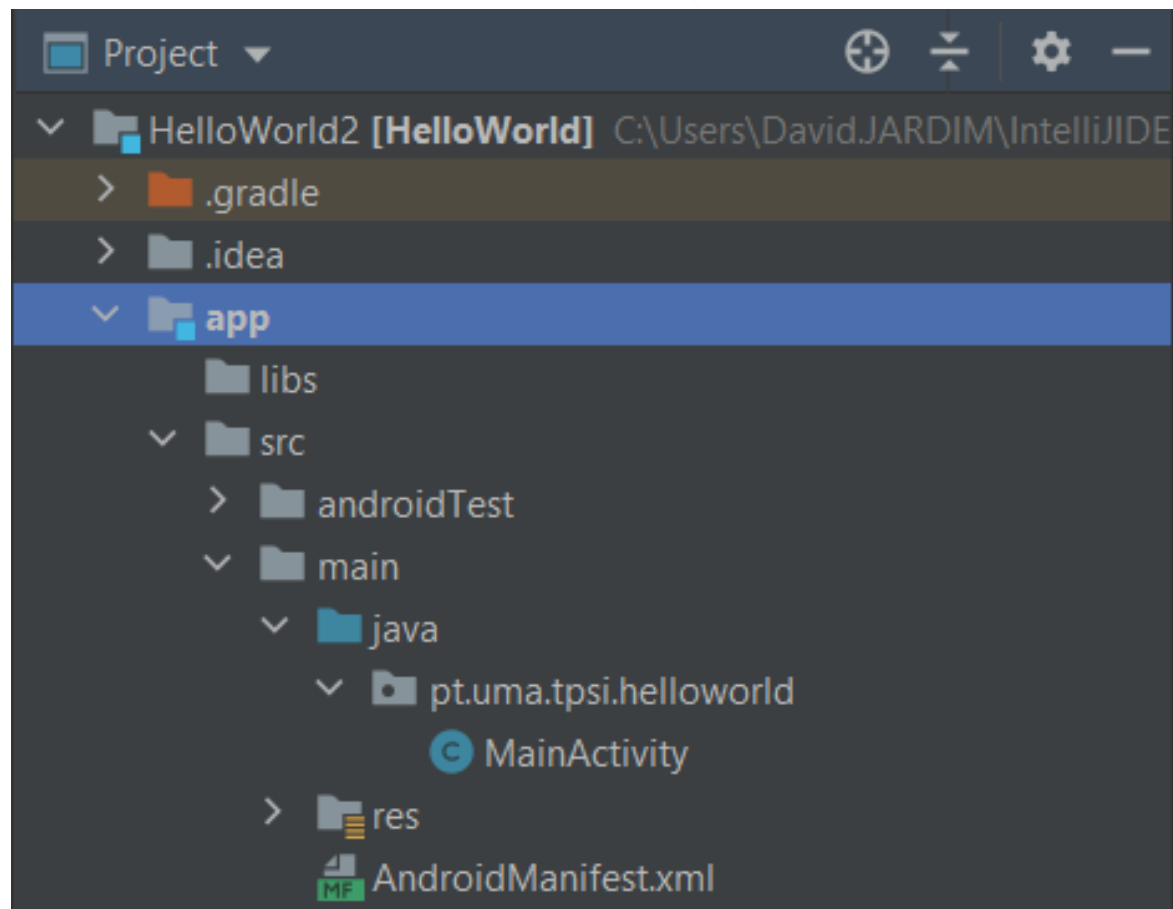


Criação de
uma
Aplicação
Android

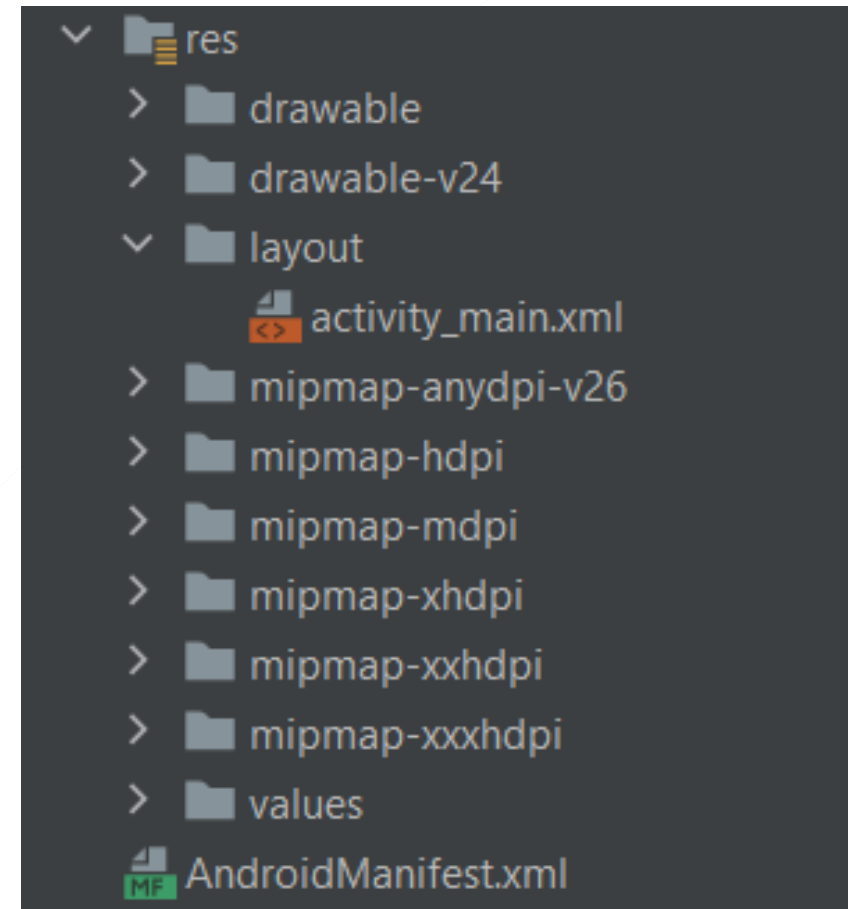


Criação de uma Aplicação Android

Código fonte



Layouts



Ficheiro AndroidManifest.xml

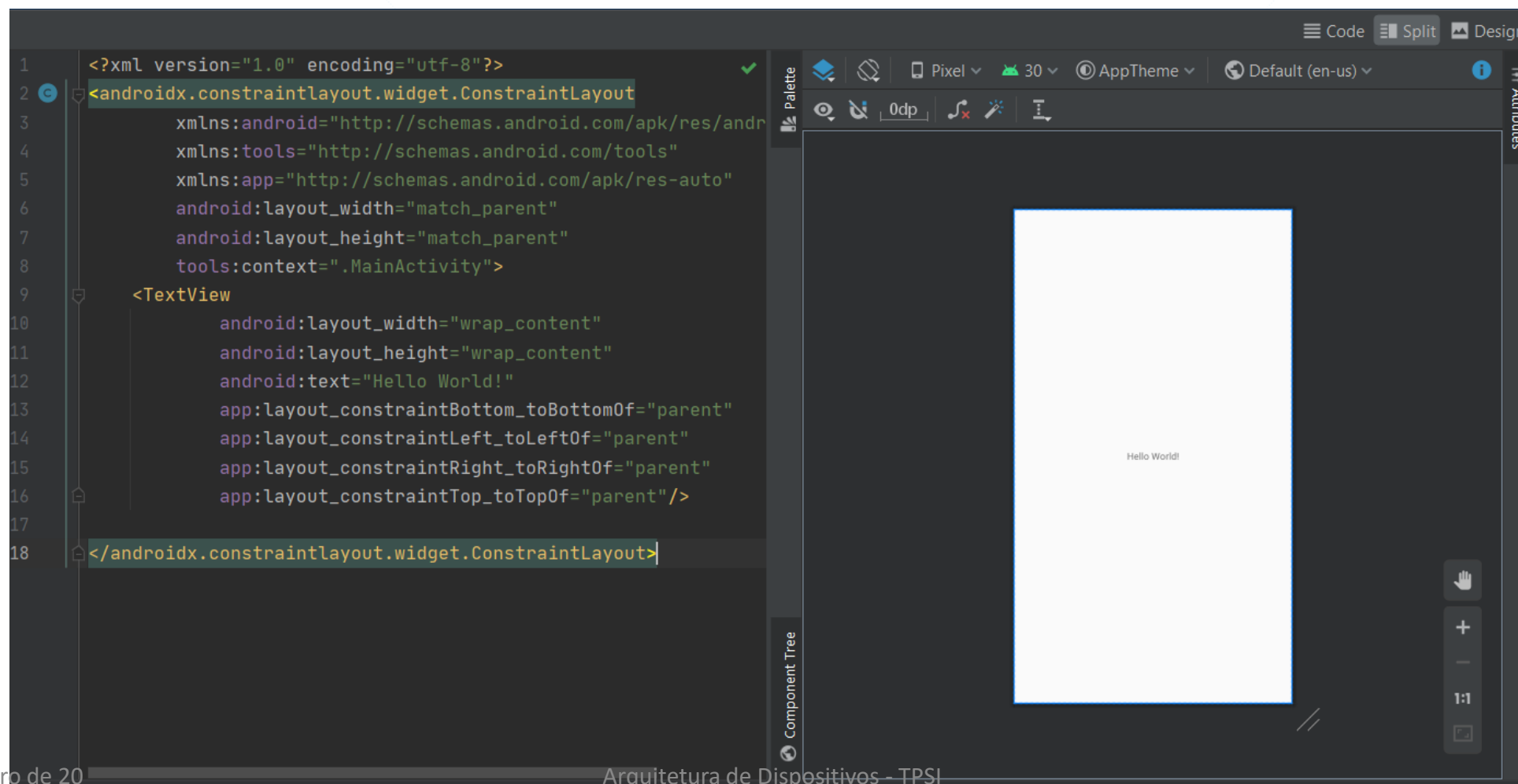
- Este ficheiro descreve informação essencial a três componentes:
 - Android build tools
 - Android OS
 - Google Play

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="pt.uma.tpsi.helloworld">

    <application
        android:allowBackup="true"
        android:icon="@mipmap/ic_launcher"
        android:label="HelloWorld"
        android:roundIcon="@mipmap/ic_launcher_round"
        android:supportsRtl="true"
        android:theme="@style/AppTheme">
        <activity android:name=".MainActivity">
            <intent-filter>
                <action android:name="android.intent.action.MAIN"/>

                <category android:name="android.intent.category.LAUNCHER"/>
            </intent-filter>
        </activity>
    </application>
    <uses-permission android:name="android.permission.SEND_SMS"/>
</manifest>
```

Ficheiro layout



Como definir o layout a utilizar?

```
public class MainActivity extends AppCompatActivity {  
  
    @Override  
    protected void onCreate(Bundle savedInstanceState) {  
        super.onCreate(savedInstanceState);  
        setContentView(R.layout.activity_main);  
    }  
}
```


Como definir um *listener* para um botão?

```
public class MainActivity extends AppCompatActivity {  
  
    @Override  
    protected void onCreate(Bundle savedInstanceState) {  
        super.onCreate(savedInstanceState);  
        setContentView(R.layout.activity_main);  
  
        Button bt = findViewById(R.id.buttonTest);  
        bt.setOnClickListener(new View.OnClickListener() {  
            @Override  
            public void onClick(View v) {  
                System.out.println("Hello World!");  
            }  
        });  
    }  
}
```

Criar uma referência para o botão

Definição do *listener* onClick

